

**ECRAN DE VISUALISATION SUSCEPTIBLE D'ETRE SOUMIS A UNE
PROCEDURE DE DEFILEMENT**

DESCRIPTION

5 DOMAINE TECHNIQUE

La présente invention concerne un procédé d'affichage d'un document, par exemple une page HTML ("Hypertext mark-up language"), sur un écran de visualisation susceptible d'être soumis à une procédure de défilement ("scrolling"), par exemple sur un écran de télévision numérique.

ETAT DE LA TECHNIQUE ANTERIEURE

Le domaine de l'invention est notamment celui de l'affichage de pages HTML, comme décrit dans la demande de brevet EP 1 164 499.

Une version de la norme HTML ("HTML 4.01 Specification W3C Recommendation 24 décembre 1999") peut être trouvée à l'adresse Internet suivante <http://www.w3.org/TR/1999/REC-htMT401-1999> 1224

Une page HTML est un document qui peut être interprété par un programme de lecture pour produire une sortie visuelle, et éventuellement audio, sur un moniteur d'ordinateur ou sur un écran de télévision. Une page HTML présentée sur un écran est constituée habituellement d'un document principal et de documents secondaires pouvant contenir des éléments graphiques, sonores ou du code source.

Afin de tenir compte de cette page HTML, ces éléments sont chargés, stockés en mémoire et traités par un moteur HTML. La représentation de cette

page est stockée dans une mémoire graphique pour affichage.

Un moteur HTML est une application interactive permettant de visualiser des documents
5 hypertexte et multimédia accessibles au travers du réseau Internet. Il permet également d'envoyer et de recevoir du courrier électronique et de consulter des serveurs dits "news".

Dans le domaine de la télévision numérique
10 une telle application interactive est pilotée par l'utilisateur au moyen d'une télécommande ou à l'aide d'un clavier infrarouge. L'application peut, en effet, intégrer une interface de substitution permettant à l'utilisateur d'effectuer toutes les saisies de texte
15 ou autres avec seulement la télécommande. L'affichage des documents est réalisé sur l'écran.

La récupération des documents sur le réseau Internet s'effectue par l'intermédiaire de requêtes sur des serveurs particuliers. Dans le domaine de la
20 télévision numérique cette opération peut être asymétrique. En effet, la requête et le document ne transitent pas nécessairement par le même milieu : La requête peut être effectuée par le réseau téléphonique via un modem, et le document revenir soit par
25 l'antenne, c'est-à-dire par émission du satellite, soit par le modem. L'utilisateur peut alors choisir le type de retour.

Le contenu d'une page HTML peut excéder l'espace disponible sur le moniteur ou sur l'écran de
30 télévision. Dans ce cas le moniteur ou l'écran de télévision n'affiche qu'une partie de la page HTML.

L'utilisateur peut alors faire défiler l'affichage pour avoir un aperçu des parties restantes de la page HTML.

Après défilement, la nouvelle partie visible du document HTML doit être affichée. Chaque
5 élément graphique en intersection avec la nouvelle partie visible doit être dessiné ou redessiné.

Le défilement d'un document est couramment réalisé de plusieurs façons. La manière la plus simple est l'affichage sans préparation préalable dans une
10 mémoire graphique tampon, c'est à dire en dessinant directement dans la fenêtre affichée à l'écran. En cas de défilement du document, le contenu de la fenêtre restant toujours visible est déplacé par une copie massive à sa nouvelle position dans la fenêtre. La
15 bande restante, nouvellement apparue, est affichée en redessinant individuellement chaque objet graphique. Ce procédé a comme inconvénient, lorsque le système graphique est lent, de voir se composer progressivement le dessin de chacun des objets graphiques.

20 Pour remédier à ce problème, une solution utilisée est de préparer la partie du document à afficher dans une mémoire graphique tampon comme par exemple un pixmap et d'afficher en une fois le contenu de ce pixmap dans la fenêtre visible. Ici deux
25 techniques sont utilisées. La première consiste à ne couvrir par cette mémoire tampon que la zone visible du document. Dans ce cas, lorsqu'un défilement est demandé la nouvelle portion visible du document est préparée dans la mémoire tampon et lorsque celle-ci est prête,
30 elle est affichée (technique appelée "double-buffering"). La seconde consiste à couvrir

l'intégralité du document avec une mémoire graphique tampon, de préparer l'intégralité du document et de l'afficher au fur et à mesure des besoins.

Les problèmes majeurs de ces deux techniques sont que, pour la première, l'affichage est préparé sur demande explicite de défilement. Par conséquent, entre la demande et le défilement effectif il peut se passer un certain temps désagréable pour l'utilisateur, le défilement n'étant pas fluide. Pour la seconde, si le problème de temps d'attente n'existe pas, elle peut nécessiter une grande quantité de mémoire graphique puisque celle-ci est liée à la taille de documents. Sur le réseau internet, la taille des documents n'est pas limitée et la quantité de mémoire graphique nécessaire pour utiliser cette seconde technique peut être considérable.

Une telle solution n'est pas, non plus, adaptée à l'affichage d'un décodeur parce que celui-ci nécessite une mémoire importante pour stocker les données pixmap qui sont calculées avant l'affichage. Or, la capacité mémoire d'un décodeur de télévision est limitée. Elle est par exemple de l'ordre de 2 mégaoctets de mémoire graphique. Elle est donc nettement plus limitée que celle d'un ordinateur qui peut être de 32 mégaoctets, ou 64 mégaoctets...

L'invention a pour objectif de résoudre un tel problème, c'est-à-dire d'offrir un défilement fluide d'un document dans un contexte de mémoire limitée, ce document pouvant par exemple être une page HTML.

EXPOSÉ DE L'INVENTION

La présente invention concerne un procédé d'affichage d'un document sur un écran de visualisation susceptible d'être soumis à une procédure de
5 défilement, caractérisé en ce qu'il comprend les étapes suivantes :

- une étape d'attribution au document d'une quantité de mémoire graphique pour créer une mémoire tampon de la partie visible du document et des zones
10 les plus proches de cette partie visible appelées "bandes d'anticipation",
- une étape de calcul et de découpage en pixmaps de cette mémoire en fonction de la taille du document, de la partie visible, et de celles des bandes
15 d'anticipation,
- une étape, de positionnement relatif de ces pixmaps par rapport au document complet et sa partie visible,
- une étape, réalisable en tâche de fond,
20 de remplissage du contenu des pixmaps avec un système de priorité en fonction de la proximité du pixmap par rapport à la zone visible,
- lorsque le document est soumis à une procédure d'affichage ou à un défilement, une étape de
25 recopie du contenu des pixmaps dans la fenêtre d'affichage avec préalablement si nécessaire une étape forçant la mise à jour des pixmaps concernés par l'affichage si l'étape précédente ne l'a pas terminée,
- et retour à l'étape de positionnement
30 relatif des pixmaps par rapport au document en fonction de la nouvelle position de la partie visible.

Avantageusement en cas de défilement horizontal et vertical, les bandes d'anticipation comportent au minimum une colonne de pixmaps à droite et à gauche de la fenêtre visible ainsi qu'une ligne de
5 pixmaps en bas et en haut, sauf dans le cas où la fenêtre visible approche le bord du document. Les pixmaps sont découpés en rectangles qui sont dessinés successivement à chaque appel d'une tache de fond.

Le remplissage d'un pixmap s'il est de
10 grande taille peut être long et bloquer le système le temps de l'opération. Le remplissage des pixmaps par rectangles de petites tailles permet de diluer l'opération de remplissage parmi les autres traitements du système, et pouvoir répondre rapidement à
15 l'utilisateur si nécessaire.

La tache de fond a également pour fonction de construire la zone d'anticipation.

A chaque appel de cette tache de fond, le processus est le suivant :

- 20 - réorganisation éventuelle des pixmaps si un défilement a été effectué,
- s'il n'y a pas eu de repositionnement des pixmaps, dessin du premier rectangle d'un pixmap déterminé en fonction d'un critère d'éloignement de la
25 zone visible du document

Avantageusement le procédé de l'invention utilise un mécanisme de synchronisation permettant le forçage éventuel des données à afficher dans les pixmaps.

Un dessin immédiat est réalisé dans deux cas :

- lorsqu'un événement "expose" oblige à dessiner une partie de la fenêtre d'affichage alors que
5 cette partie n'est pas encore dessinée dans les bandes d'anticipation,

- lorsqu'un élément du document est modifié graphiquement dans la fenêtre d'affichage.

Le procédé de l'invention peut être
10 notamment utilisé pour l'affichage d'un document HTML, et/ou pour un décodeur de télévision numérique.

BRÈVE DESCRIPTION DES DESSINS

La figure 1 illustre la visualisation de la zone d'anticipation selon le procédé de l'invention.

15 La figure 2 illustre un exemple de déplacement de la zone d'anticipation.

La figure 3 illustre le mécanisme de synchronisation, lorsque une partie des pixmaps n'est pas remplie et qu'un affichage est demandé.

20 La figure 4 illustre la constitution d'une arborescence de sous-fenêtres dans un exemple de mise en œuvre du procédé de l'invention.

La figure 5 illustre l'affichage de l'arborescence représentée sur la figure 4.

25 Les figures 6 et 7 illustrent, chacune, un document HTML après mise en page selon le procédé de l'invention ; la largeur complète du document pouvant être couverte avec les pixmaps pour la figure 7, alors que pour la figure 6 ce n'est pas possible.

EXPOSÉ DÉTAILLÉ DE MODES DE RÉALISATION PARTICULIERS

Le procédé, selon l'invention, d'affichage d'un document, par exemple une page HTML, sur un écran de visualisation susceptible d'être soumis à une
5 procédure de défilement consiste à découper ce document en zones pouvant être couvertes par des pixmaps dont la taille est fonction de la taille de la zone de visualisation, et à prévoir une zone d'anticipation partie du document réellement couverte par des pixmaps.

10 Cette zone d'anticipation forme une matrice de pixmaps, couvrant du document, la partie visible et les bandes les plus proches autour de cette partie visible, de manière à préparer le contenu de la partie visible et les zones prochainement visibles en cas de
15 défilement du document.

La zone d'anticipation a comme finalité d'améliorer le rendu visuel des opérations de redessin d'une partie du document. En effet, dans la plupart des cas, une copie de la zone d'anticipation dans la
20 fenêtre d'affichage s'avère suffisante.

La figure 1 représente une telle zone d'anticipation 10, formée ici de 18 lignes et 12 colonnes de pixmaps 13, la fenêtre visible 11 où le document 12 est visualisé.

25 On constate l'inclusion hiérarchique des trois zones ainsi formées, le document 12 incluant la zone d'anticipation 10, cette dernière contenant la fenêtre visible 11 du document. La zone d'anticipation peut très bien recouvrir en largeur ou en hauteur le
30 document si l'espace disponible est suffisant, comme le montre l'illustration de la figure 7.

La figure 2 montre l'évolution de la zone d'anticipation 10 suite à une opération de défilement de l'affichage du document 12 vers le bas. Cette opération de défilement entraîne un déplacement de la première ligne en bas de la zone d'anticipation. Cette ligne doit alors être redessinée.

La zone d'anticipation comporte au minimum une colonne de zones pixmap à droite et à gauche de la fenêtre visible ainsi qu'une ligne de zones pixmap en bas et en haut, sauf dans le cas où la fenêtre visible 11 approche le bord du document 12. Si cette règle n'est pas respectée à la suite d'un défilement, la zone d'anticipation se déplace de manière à rétablir celle-ci.

Pour ne pas risquer de bloquer l'application pendant plusieurs secondes, le dessin de la zone d'anticipation, suite à sa création ou à un défilement, est tel qu'une tâche de fond effectue ce dessin en plusieurs étapes : les zones pixmaps sont 20 formées en rectangles qui sont dessinés successivement à chaque appel de la tâche de fond.

Cette tâche de fond a également pour fonction de construire la zone d'anticipation.

A chaque appel de cette tâche de fond, le processus est le suivant :

- déplacement si nécessaire de la zone d'anticipation par permutation de lignes ou de colonnes en pixmaps,
- si la zone d'anticipation est 30 correctement positionnée, dessin du premier rectangle d'un pixmap de la zone d'anticipation.

Si le pixmap à afficher n'est pas prêt au moment de l'affichage, un mécanisme de synchronisation permet de forcer le rafraîchissement des données à afficher dans les pixmaps.

5 Un dessin immédiat est réalisé dans deux cas :

- lorsqu'un événement "expose" oblige à dessiner une partie de la fenêtre d'affichage alors que cette partie n'est pas encore dessinée dans la zone d'anticipation,

10 - lorsqu'un élément du document est modifié graphiquement dans la fenêtre d'affichage.

Ces deux cas correspondent respectivement à un afflux de nouvelles données concernant une image et au traitement du focus qui modifie le dessin d'un

15 élément de l'image.

Dans ces deux cas tous les rectangles en attente ayant une intersection commune avec la zone à dessiner sont dessinés. La figure 3 représente les

20 éléments qui composent alors la zone d'anticipation :

- les rectangles déjà dessinés 20,
- les rectangles non dessinés 21,
- la fenêtre d'affichage 22,
- le rectangle 23 à dessiner suite à un

25 événement "expose",

- les rectangles non dessinés 24 qui sont en interaction avec le rectangle 23, le dessin de leur contenu étant forcé par la synchronisation.

Mise en œuvre du procédé de l'invention dans une application HTML

On va, à présent, considérer un exemple de mise en œuvre du procédé de l'invention, dans une application HTML dans un décodeur.

La première opération consiste à répartir la mémoire graphique utilisable dans une arborescence de sous-fenêtres (ou "frames" HTML).

La quantité de mémoire graphique disponible pour créer les pixmaps est limitée au niveau du décodeur. Pour garantir un bon fonctionnement des autres applications fonctionnant en même temps que le moteur HTML, seule une partie de cette mémoire est utilisée par le moteur HTML pour créer les zones d'anticipation. La quantité de mémoire disponible est définie au lancement du moteur HTML. Dans le cas où plusieurs documents déroulants sont visibles simultanément à l'écran (cas des sous-fenêtres ou "frames" HTML), il faut que chaque document puisse profiter du mécanisme de déroulement fluide que constitue le procédé de l'invention. La quantité de mémoire disponible doit donc être répartie sur tous les documents visibles simultanément.

Le procédé de l'invention ne concerne que des documents déroulants. Les documents déclarant des sous-fenêtres HTML ne peuvent jamais être déroulants. Seuls les documents feuilles d'une arborescence de sous-fenêtres sont susceptibles d'être déroulants, et sont donc utilisateurs potentiels du procédé de l'invention. Ainsi, la somme des parties visibles des documents HTML susceptible d'être déroulante couvre

exactement la surface réservée à l'affichage du moteur HTML. Cette propriété est utilisée pour garantir une utilisation du procédé de l'invention pour chaque document feuille.

5 Le moteur HTML répartit proportionnellement la mémoire graphique, en fonction de la surface de chaque sous-fenêtre :

- M représente la quantité de mémoire graphique réservée au moteur HTML,

10 - W_m & H_m : la taille en pixels de la fenêtre d'affichage du moteur HTML,

- W_f & H_f : la taille en pixels d'une sous-fenêtre quelconque.

La quantité de mémoire graphique utilisable
15 par cette sous-fenêtre pour créer ses zones d'anticipation est proportionnelle à la surface :

$$M_f = M * (W_f * H_f) / (W_m * H_m)$$

Un exemple d'arborescence de sous-fenêtre, dans laquelle chaque case constitue un document HTML,
20 est illustré sur la figure 4. On a ainsi un document racine, par exemple de 600 x 400 pixels, dont découle un document Frame 1, par exemple de 600 x 100 pixels, et un document Frame 2, par exemple de 600 x 300 pixels, dont découle un document Frame 2-1, par exemple
25 de 200 x 300 pixels et un document Frame 2-2, par exemple de 400 x 300 pixels.

L'affichage d'une telle arborescence donne le résultat illustré sur la figure 5.

Dans un tel exemple la répartition de la
30 mémoire graphique peut être réalisée de la façon suivante, avec, $M = 1.920$ Koctets :

- Document racine (déclaration de frames)=
0 octet
- "Frame 1" (document feuille)= 480 Koctets
- "Frame 2" (déclaration de "frames")= 0
5 octet
- "Frame 2.1" (document feuille)= 480
Koctets
- "Frame 2.2" (document feuille)= 960
Koctets

10 L'opération suivante est un problème de
détermination de granularité des pixmaps : pour une
quantité de mémoire graphique disponible pour un
document il faut déterminer quelle est la taille et la
granularité des pixmaps qui va permettre le bon
15 fonctionnement du système. L'objectif est de garantir
au moins un pas de défilement, d'anticipation du
mouvement de part et d'autre de la zone visible, et
d'avoir au moins une rangée de pixmaps pouvant être
déplacée d'un côté ou de l'autre.

20 La figure 6 représente un document HTML 12
complet après la mise en page, la partie couverte par
les pixmaps 13 et la partie visible 11. Ce document 12
est susceptible à la fois d'avoir un défilement
horizontal et un défilement vertical.

25 Sur cette figure sont représentés les
paramètres suivants :

- *Wd* & *Hd* : largeur et hauteur en pixels du
document complet après mise en page,
- *Wf* & *Hf* : largeur et hauteur en pixels de
30 la partie visible du document (taille de la sous-
fenêtre),

- Wp & Hp : largeur et hauteur en pixels d'un pixmap,

Il existe d'autres paramètres non représentés sur cette figure :

5 - Nx & Ny : nombre de pixmaps en horizontal ou en vertical,

 - Px & Py : bande d'anticipation garantie disponible en horizontal et en vertical de part et d'autre de la zone visible (cette bande d'anticipation
10 correspond, au minimum, au pas de défilement),

 - T : taille d'un pixel en octets (fonction du système de codage des couleurs),

 - Mf : quantité de mémoire graphique disponible pour les pixmaps associés au document.

15 On considère tout d'abord que le document
12 peut être déroulé horizontalement et verticalement
et que la quantité de mémoire graphique est
insuffisante pour couvrir l'intégralité de celui-ci
avec des pixmaps aussi bien en horizontal qu'en
20 vertical.

Pour pouvoir garantir à la fois la permutation des pixmaps et l'existence d'une bande d'anticipation garantie de chaque côté, on doit respecter les inégalités suivantes :

25 - $2 Px + Wf \leq Wp (Nx-1)+1$ (1)

avec $Nx > 1$

- $2 Py + Hf \leq Hp (Ny-1)+1$ (2)

avec $Ny > 1$

La limitation mémoire impose également une contrainte. La somme des pixmaps ne doit pas dépasser la quantité de mémoire réservée :

$$Mf \geq Nx * Ny * Wp * Hp * T \quad (3)$$

5 Le nombre d'équations est encore insuffisant pour déterminer le nombre et la taille des pixmaps. Aussi, on opte pour certaines conditions particulières :

10 - Px , la zone d'anticipation horizontale garantie, est définie comme égale au pas de défilement horizontal.

- Py , la zone d'anticipation verticale garantie, est définie comme égale au pas de défilement vertical.

15 - $Px = \alpha Wf$: le pas horizontal de défilement est proportionnel à la largeur d'affichage du document.

- $Py = \alpha Hf$: le pas vertical de défilement est proportionnel à la hauteur d'affichage du document avec $0 < \alpha \leq 1$, α étant une constante du moteur HTML (contrainte garantissant que l'intégralité du document peut être consulté par un déroulement pas à pas).

20 - $(Wp * Nx) * Hf = (Hp * Ny) * Wf$ (4) : les dimensions de la zone d'anticipation sont proportionnelles aux dimensions de la zone d'affichage.

25 On détermine la taille et le nombre de pixmaps en horizontal en utilisant les équations (3) et (4) et en considérant que la capacité maximum de la mémoire graphique doit être utilisée. On obtient le
30 résultat suivant :

$$W_p * Nx = ((Mf * Wf) / (Hf * T))^{1/2} \quad (5)$$

où $W_p * Nx$ correspond à la largeur en pixel de la zone d'anticipation.

En utilisant les équations (1) et (5), on obtient la largeur maximum d'un pixmap :

$$W_{pmax} = ((Mf * Wf) / (Hf * T))^{1/2} + 1 - (2 \alpha + 1) Wf$$

En divisant la largeur de la zone d'anticipation ($W_p * Nx$) de (5) par la largeur maximum d'un pixmap W_{pmax} et en arrondissant à la valeur entière supérieure, on obtient le nombre de zones pixmap minimum en largeur Nx_{min} .

Pour toute les valeurs de $Nx \geq Nx_{min}$, le bon fonctionnement du procédé de l'invention est assuré. Dans la mise en œuvre du moteur HTML, c'est la valeur Nx_{min} qui est retenue pour le découpage. En effet, au niveau du décodeur, le nombre de pixmaps pouvant être créé est limité. On garantit ainsi au maximum la possibilité d'utiliser le procédé de l'invention et on évite donc d'atteindre une telle valeur critique.

La largeur réelle d'un pixmap W_p est obtenue en divisant la largeur de la zone d'anticipation ($W_p * Nx$) de (5) par le nombre de pixmaps retenu Nx_{min} arrondi au nombre entier inférieur.

Dans l'exemple de la "Frame 2.2", en prenant un coefficient α à 10% et $T = 4$ octets, les résultats obtenus sont les suivants :

- $Nx = 7$
- $W_p = 80$ pixels

La détermination de la taille et du nombre de pixmaps en vertical est réalisée de la même façon qu'en horizontal.

Dans l'exemple considéré, les résultats obtenus pour "Frame 2.2" sont les suivants :

- $N_y = 7$
- $H_p = 60$ pixels

Avec cette méthode de calcul, la granularité des pixmaps est toujours la même en horizontal et en vertical.

Dans la mise en œuvre du décodeur, il faut tenir compte d'un paramètre supplémentaire : les pixmaps créés doivent toujours avoir des dimensions multiples de 8 pixels. Par conséquent, après avoir déterminé la taille de chaque pixmap et le nombre de pixmaps dans une direction, en arrondissant cette taille au multiple de 8 inférieur, on vérifie que l'équation (1) ou (2) selon la première direction traitée) est toujours validée. Si ce n'est plus le cas, il faut augmenter de 1 la granularité des pixmaps et recalculer la nouvelle taille d'un pixmap.

Avec ces arrondis, une partie de la mémoire réservée pour le calcul dans une direction peut n'être plus totalement utilisée. Ce trop plein de mémoire est alors reporté sur l'autre direction avant de calculer la taille et le nombre de pixmaps. Dans ces nouvelles conditions, la granularité horizontale et verticale peuvent devenir différente.

Comme illustré sur la figure 7, dans le cas ou $W_p * N_x \geq W_d$, il est possible de couvrir toute la largeur du document avec des pixmaps. Il n'y a donc pas

besoin d'un mécanisme de permutation horizontal. La
détermination du nombre et de la granularité des
pixmap est différente. Le trop plein de mémoire
graphique disponible au niveau du défilement horizontal
5 est affecté au défilement vertical. La largeur des
pixmap est alors la largeur maximum de création d'un
pixmap dans un décodeur.

REVENDICATIONS

1. Procédé d'affichage d'un document (12) sur un écran de visualisation pouvant être soumis à une
- 5 procédure de défilement, caractérisé en ce qu'il comprend les étapes suivantes :
- une étape d'attribution au document d'une quantité de mémoire graphique pour créer une mémoire tampon de la partie visible du document et des zones

10 les plus proches de cette partie visible appelées bandes d'anticipation (10),

 - une étape de calcul et de découpage en pixmaps de cette mémoire en fonction de la taille du document, de la partie visible, et de celles des bandes

15 d'anticipation (10),

 - une étape de positionnement relatif de ces pixmaps par rapport au document complet et sa partie visible, - une étape, réalisable en tâche de fond,

20 de remplissage du contenu des pixmaps avec un système de priorité en fonction de la proximité du pixmap par rapport à la zone visible,

 - lorsque le document est soumis à une procédure d'affichage ou à un défilement, une étape de

25 recopie du contenu des pixmaps dans la fenêtre d'affichage avec préalablement si nécessaire une étape forçant la mise à jour des pixmaps concernés par l'affichage si l'étape précédente ne l'a pas terminée,

 - et retour à l'étape de positionnement

30 relatif des pixmaps par rapport aux documents en fonction de la nouvelle position de la partie visible.

2. Procédé selon la revendication 1, dans lequel les bandes d'anticipations (10) comportent au minimum une colonne de pixmaps à droite et à gauche de la fenêtre visible (11) ainsi qu'une ligne de pixmaps en bas et en haut, sauf dans le cas où la fenêtre visible (11) approche le bord du document (12).

3. Procédé selon la revendication 1, dans lequel les pixmaps (13) sont découpés en rectangles qui sont dessinés successivement à chaque appel d'une tache de fond.

4. Procédé selon la revendication 3, dans lequel la tache de fond a également pour fonction de construire les bandes d'anticipation (10).

5. Procédé selon la revendication 3, dans lequel, à chaque appel de cette tache de fond, on a :

- réorganisation éventuelle des pixmaps si un défilement a été effectué,

- s'il n'y pas eu de repositionnement des pixmaps, dessin du premier rectangle d'un pixmap déterminé en fonction d'un critère d'éloignement de la zone visible du document.

6. Procédé selon la revendication 1, qui utilise un mécanisme de synchronisation permettant le forçage éventuel des données à afficher dans les pixmaps.

7. Procédé selon la revendication 1, dans lequel un dessin immédiat est réalisé dans deux cas :

- lorsqu'un événement "expose" oblige à dessiner une partie de la fenêtre d'affichage alors que
5 cette partie n'est pas encore dessinée dans les bandes d'anticipation (10),

- ou lorsqu'un élément du document est modifié graphiquement dans la fenêtre d'affichage.

10 8. Utilisation du procédé selon l'une quelconque des revendications précédentes pour l'affichage d'un document HTML.

15 9. Utilisation du procédé selon l'une quelconque des revendications 1 à 7 dans un décodeur de télévision numérique.

1 / 3

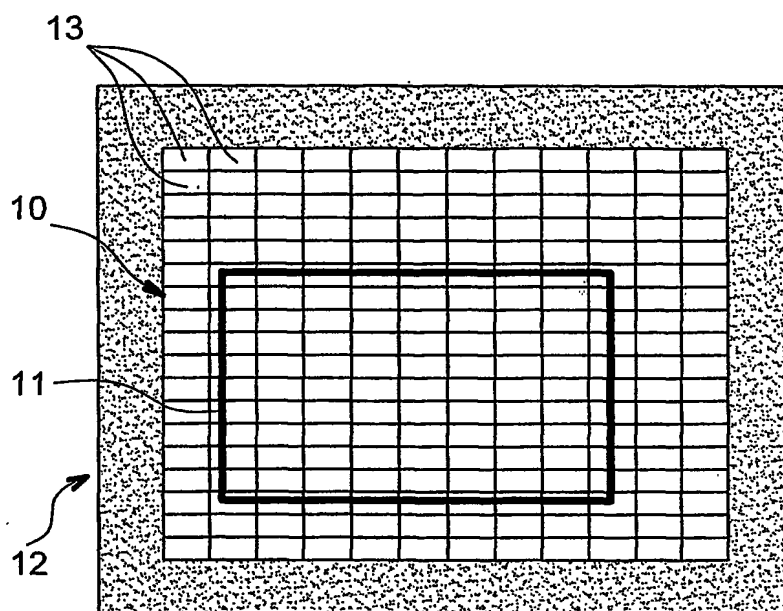


FIG. 1

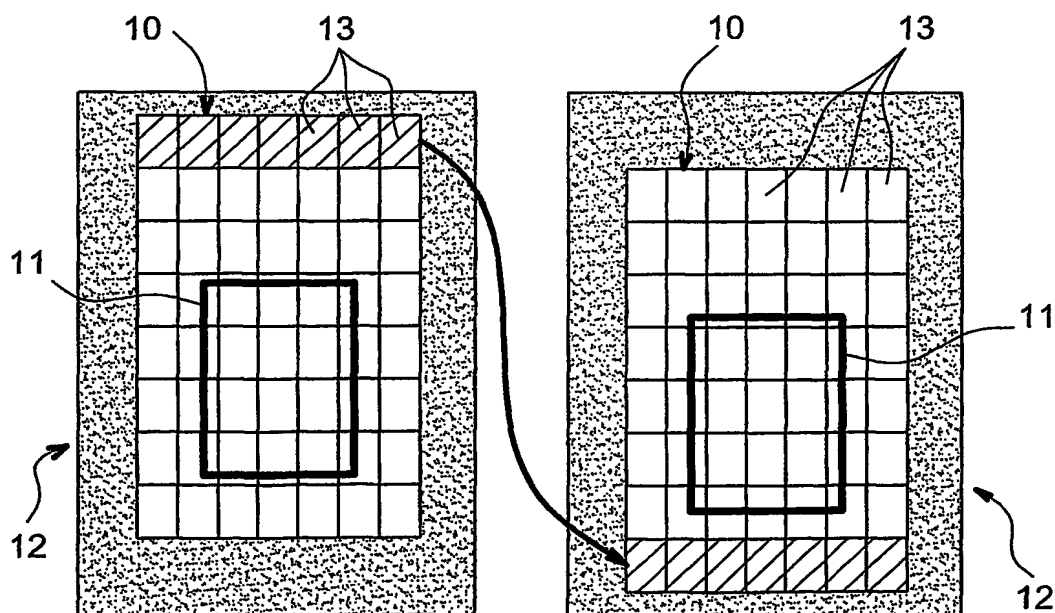


FIG. 2

2 / 3

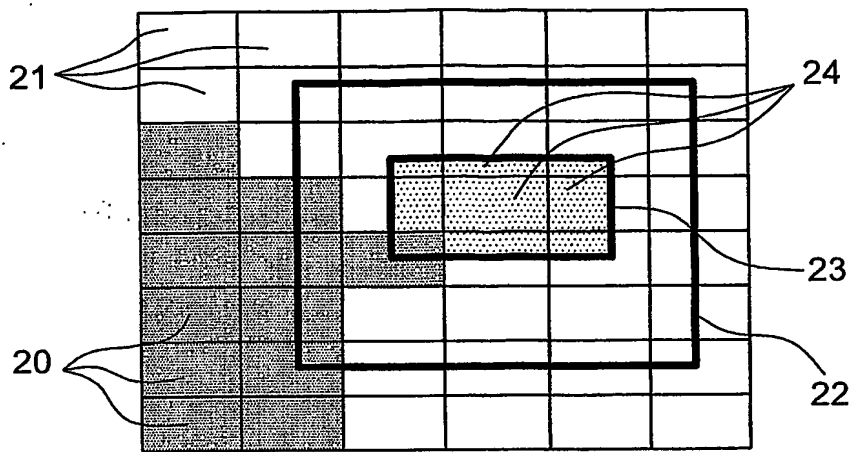


FIG. 3

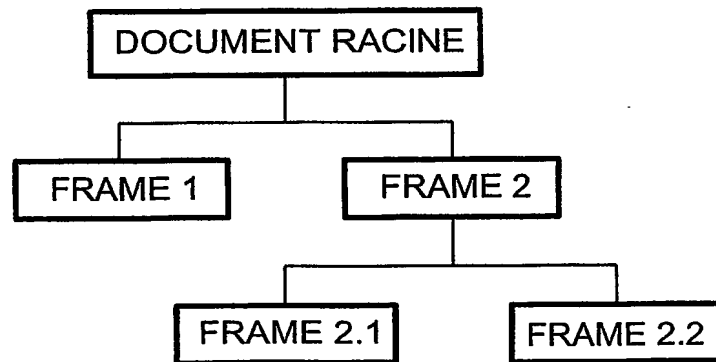


FIG. 4

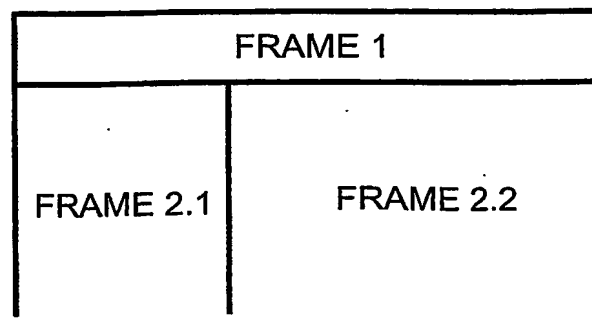


FIG. 5

3 / 3

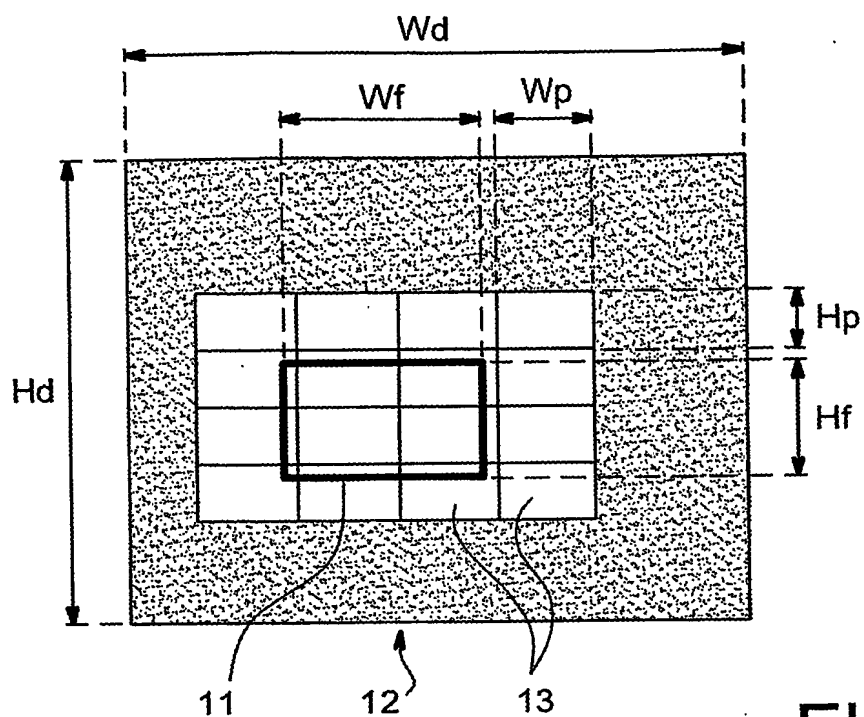


FIG. 6

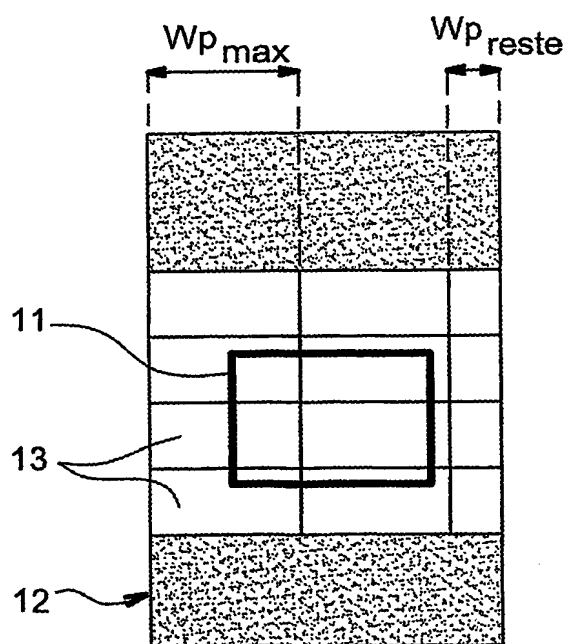


FIG. 7